### Median in a row-wise sorted Matrix

Given a row wise sorted matrix of size **R\*C** where R and C are always **odd**, find the median of the matrix.

**Example 1:**

**Input**:

R = 3, C = 3

M = [[1, 3, 5],

  [2, 6, 9],

  [3, 6, 9]]

**Output:** 5

**Explanation**: Sorting matrix elements gives

us {1,2,3,3,5,6,6,9,9}. Hence, 5 is median.

**Example 2:**

**Input:**

R = 3, C = 1

M = [[1], [2], [3]]

**Output:** 2

**Explanation**: Sorting matrix elements gives

us {1,2,3}. Hence, 2 is median.

### Java code

import java.io.\*;

import java.util.\*;

class CodingMaxima

{

public static void main(String args[])throws IOException

{

BufferedReader read = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(read.readLine());

while(t-- > 0)

{

String S[] = read.readLine().split(" ");

int R = Integer.parseInt(S[0]);

int C = Integer.parseInt(S[1]);

int matrix[][] = new int[R][C];

int c = 0;

for(int i = 0; i < R; i++){

String line[]=read.readLine().trim().split(" ");

for(int j = 0; j < C; j++){

matrix[i][j] = Integer.parseInt(line[j]);

}

}

Solution ob = new Solution();

int ans = ob.median(matrix, R, C);

System.out.println(ans);

}

}

}

class Solution {

int median(int matrix[][], int R, int C) {

// code here

ArrayList<Integer> arr=new ArrayList<Integer>();

for(int i=0;i<R;i++){

for(int j=0;j<C;j++){

arr.add(matrix[i][j]);

}

}

Collections.sort(arr);

if(arr.size()%2!=0){

return arr.get(((arr.size()+1)/2)-1);

}

else{

return (int)(arr.get(arr.size()/2-1)+arr.get(arr.size()/2))/2;

}

}

}